Software Requirements and Design Document

for

FAST Travels

Prepared by:
Muhammad Rafay i220948
Arham Khalid i221156
Ali bin Salman i220894

FAST Travels Limited

27th, November, 2024

1. Introduction

1.1 Purpose

The purpose of this Travel Management System is to provide a comprehensive and unified platform for managing various travel-related operations. It enables users to book tickets for buses, trains, and flights, as well as cargo shipments, while offering seamless management of routes, vehicles, and loyalty points. The system aims to streamline administrative tasks such as handling complaints, managing discounts, and processing payments efficiently. By integrating features like personalized travel history, user profile updates, and flexible booking modifications, the project enhances user experience and simplifies travel management. The implementation also emphasizes object-oriented principles, ensuring scalability, maintainability, and robust functionality.

1.2 Product Scope

The project, FAST Travels, is an integrated software system designed to streamline the ticketing and travel planning process for buses, trains, and airplanes. It caters to a wide audience, including the general public, university students, and frequent travelers. The system enables users to plan and book short and long journeys, through a single, unified platform, eliminating the need to rely on multiple applications or websites.
Key features will include real-time ticket availability, booking, and cancellation options across multiple transport types. Additionally, the platform will introduce loyalty programs such as a mileage points system for frequent travelers, as well as special discounts for students. FAST Travels will be built with a robust backend using Java and a secure database system based on MySQL, ensuring high performance, data integrity, and scalability as user demand increases.

1.3 Title

FAST Travels: Ticketing Solution for Comprehensive Travel

1.4 Objectives

The main objectives of FAST travels are:

- To provide a user-friendly website where clients can easily reserve tickets for trains, buses, and aircraft.

- To reduce the need for consumers to rely on multiple websites or applications by consolidating various travel options into a single, unified platform, streamlining the ticketing process.
- To implement a dependable and efficient database system that manages reservations, cancellations, and real-time ticket availability with high accuracy and reliability.
- To introduce features such as mileage points to encourage frequent travelers and enhance customer loyalty.
- To offer both long-distance and intra-city travel options, making the platform versatile for a wide range of users, including students, professionals, and casual travelers.
- To ensure the system remains scalable, secure, and responsive to the growing needs of users, with a focus on ease of use and quick processing times.

1.5 Problem Statement

Managing transportation options across buses, trains, and airplanes can be a frustrating and time-consuming task for travelers. Users are often required to visit different platforms to compare schedules, check availability, and make reservations. This fragmented process increases the likelihood of errors, missed connections, or scheduling conflicts, leading to a less-than-optimal user experience. Furthermore, frequent travelers lack a centralized system for managing their trips and reaping benefits, such as loyalty programs or discounts.
FAST Travels addresses these pain points by integrating multiple transportation services into a single, unified platform. Travelers will no longer need to juggle between different apps or websites for various modes of travel, as FAST Travels will consolidate these services in one place. The platform will provide users with up-to-date information on ticket availability, schedules, and pricing, enabling efficient and well-informed decision-making.

2. Overall Description

2.1 Product Perspective

The product FAST Travels aims to specialize already existing travel applications by unifying the platform for all types of bookings i.e train, bus and flights. In addition to this, one of the prominent features of FAST Travels is its cargo booking functionality which sets it apart from all the accessible travel booking applications in the market.

2.2 Product Functions

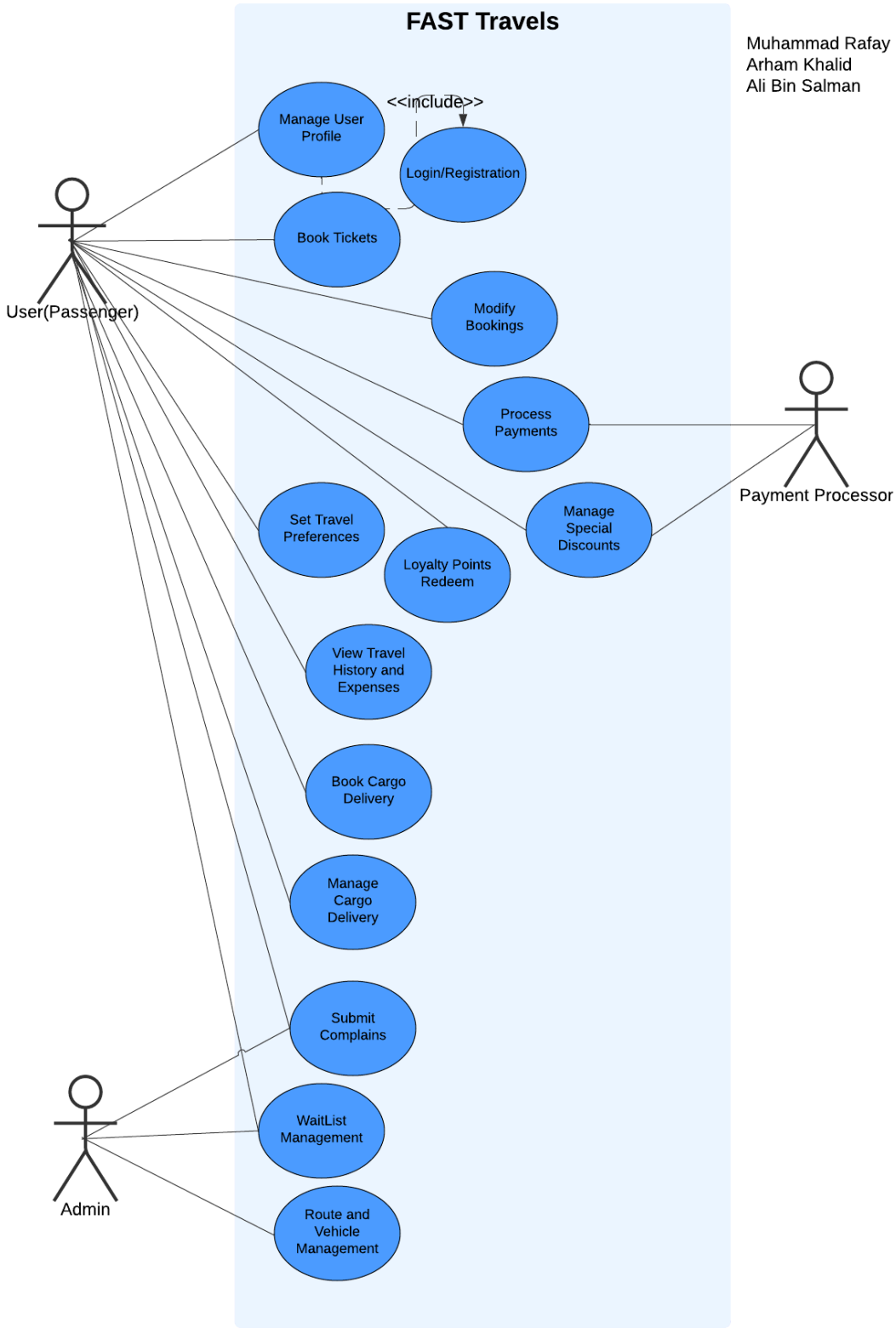
The application enables users to generate their profiles by providing valid credentials. Upon successful entry, the user can choose to generate a booking for trains, buses and flights depending upon the required dates and seats. On successful booking the user can proceed to pay with cash on arrival or with card payment. The user can also modify these bookings if he or she wishes to modify the date or cancel the whole booking. The user can also choose to make a cargo booking on selected routes. The user is provided with the option to choose a preferred travel company. For feedback and growth of the company the user can submit complaints which can be viewed by the administrators for quality insurance. The user can also earn loyalty points upon booking which can be redeemed for monetary value.

2.3 List of Use Cases

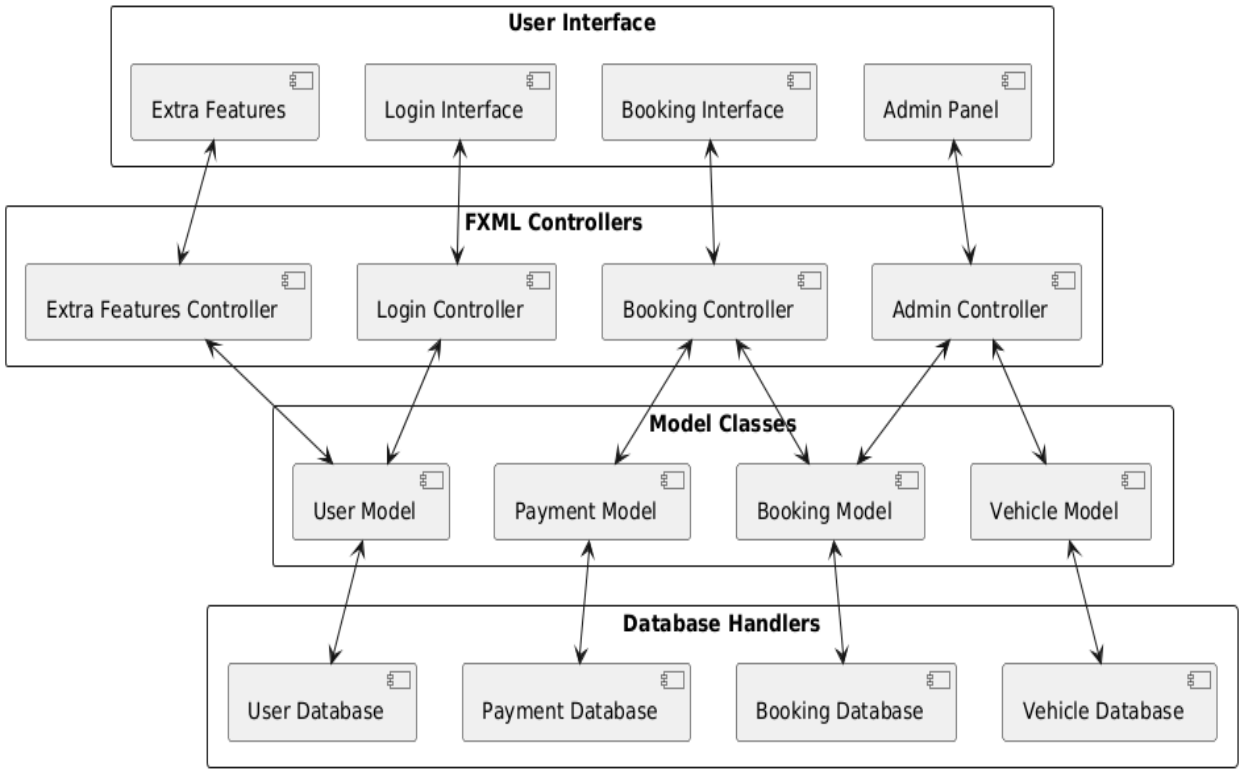
Here are the use case names extracted from the document:

1. Manage User Profile
2. Book Tickets
3. Modify Booking
4. Process Payments
5. Earn and Redeem Loyalty Points
6. Submit Complaints
7. Manage Special Discounts
8. Waitlist Management
9. Set Travel Preferences for Future Bookings
10. Book Cargo Delivery
11. Modify Cargo Booking (Cancel, Reschedule)
12. View travel history and expenses
13. Manage special discounts

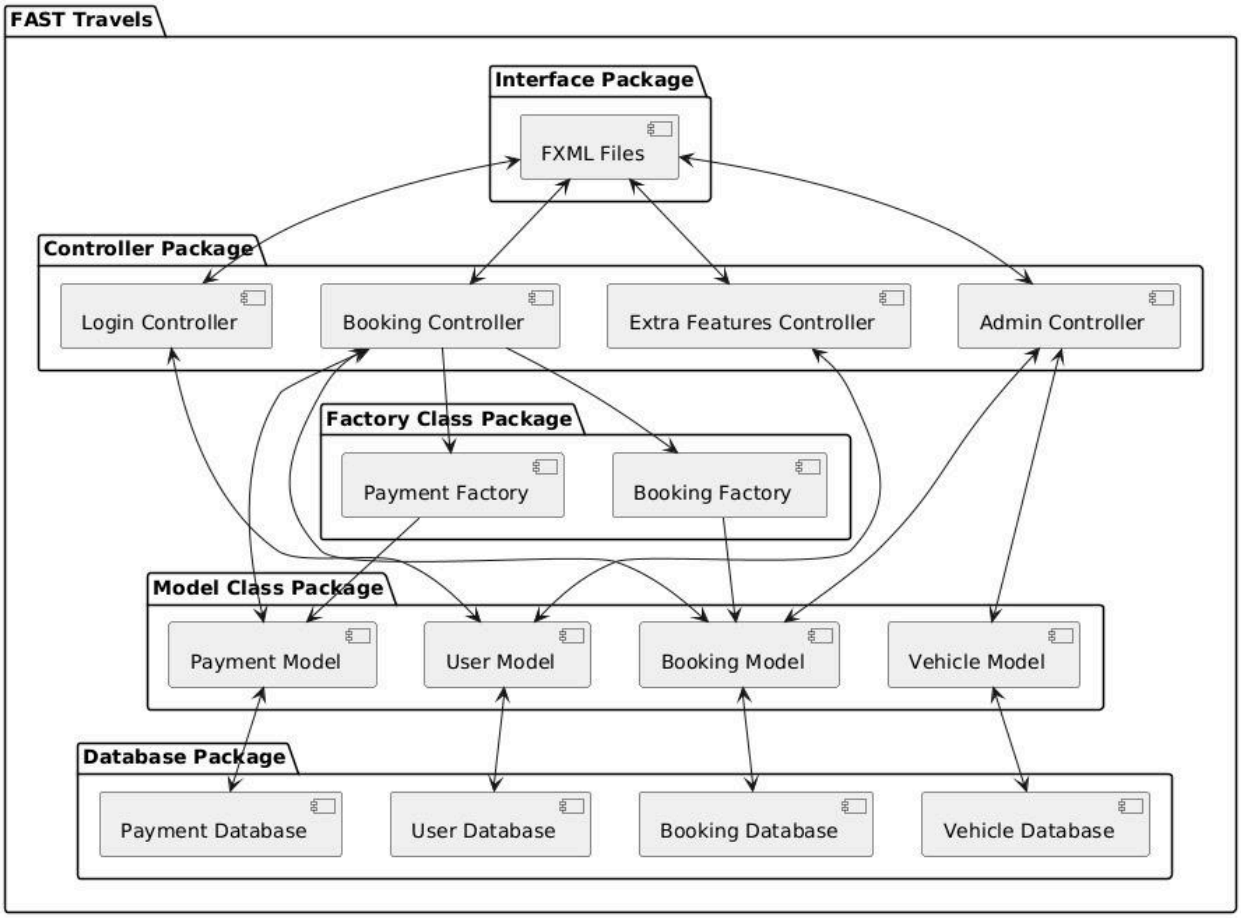
2.4 Use Case Diagram



2.4.1 Component Diagram

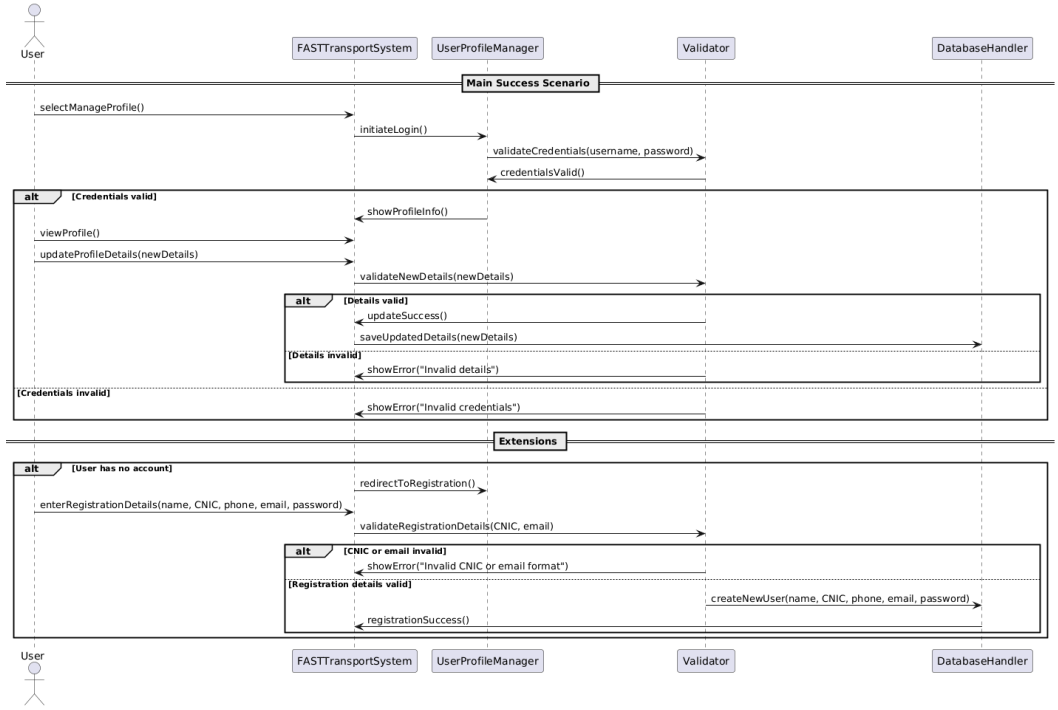


2.4.2 Package Diagram

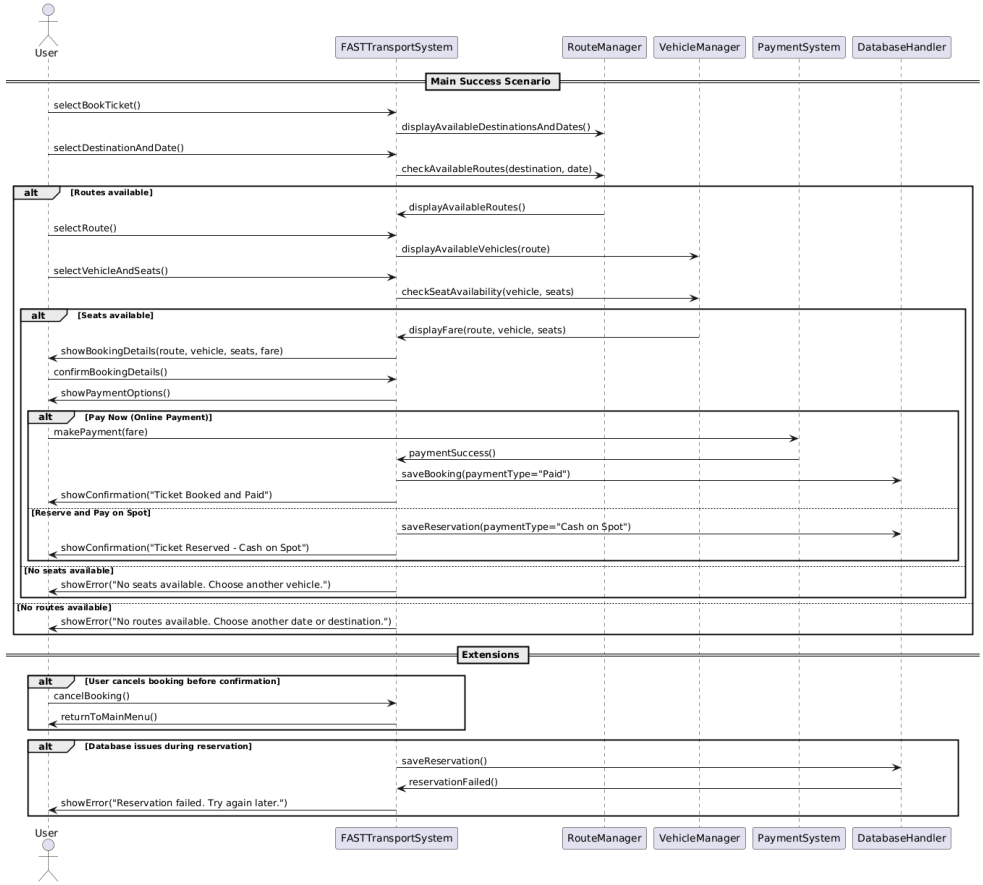


2.4.3 Major Use Cases:

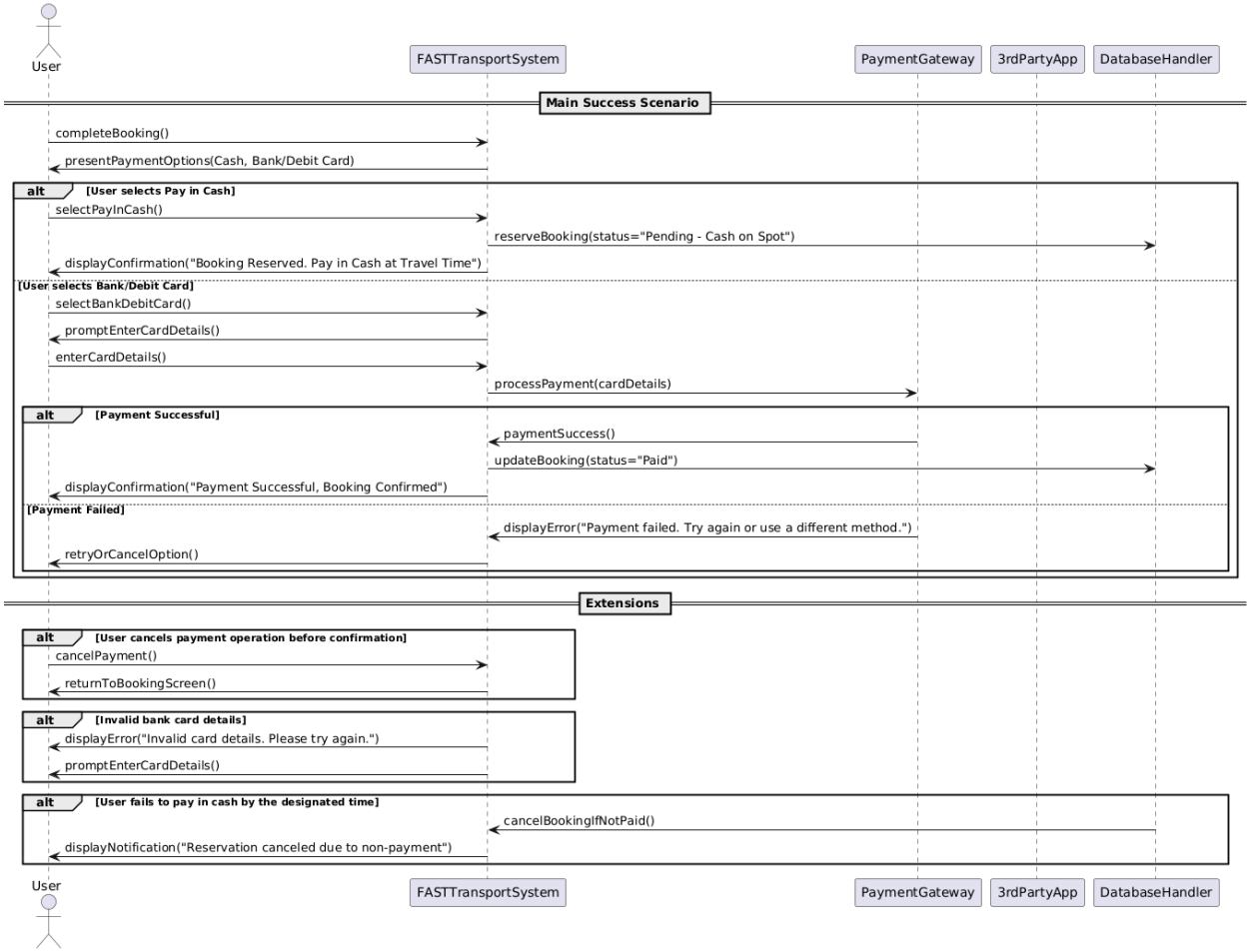
UC-01:



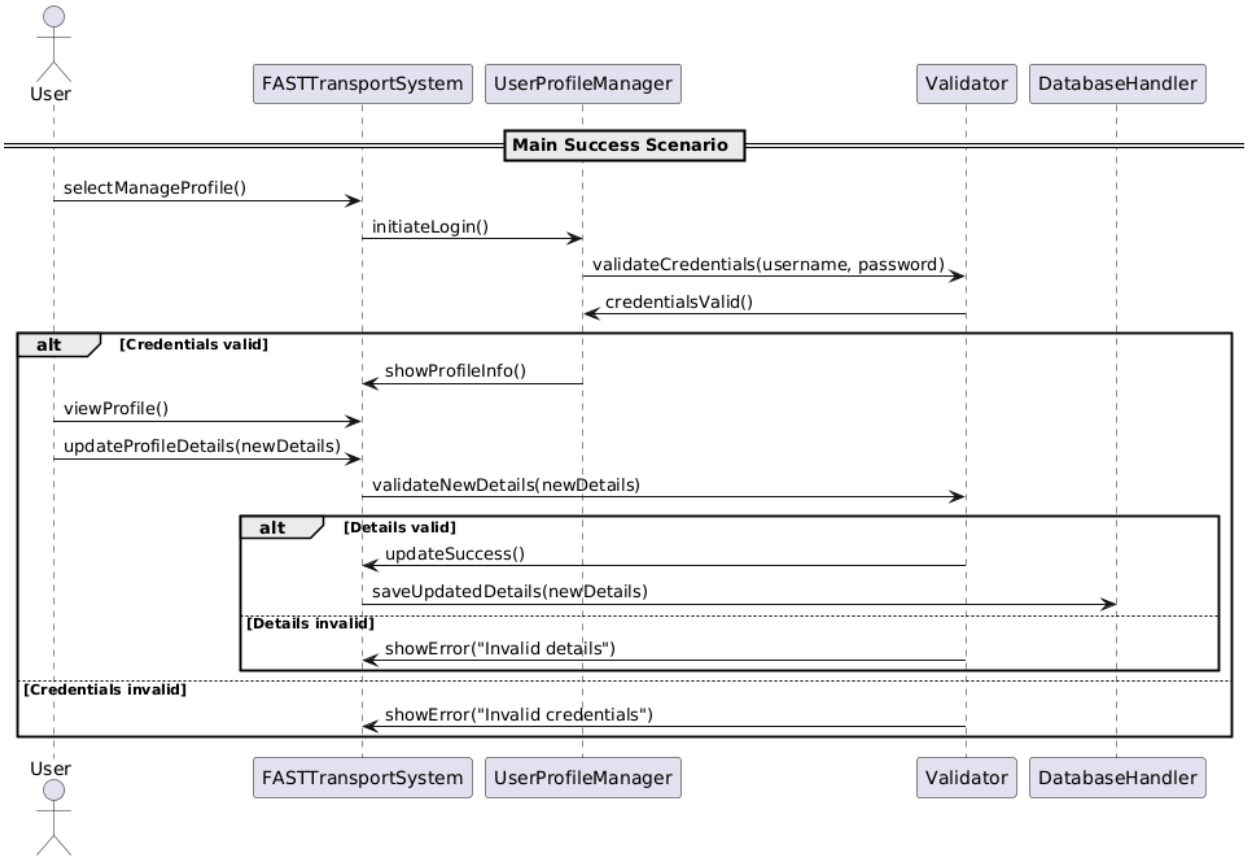
UC-02:


User
FASTTransportSystem
RouteManager
VehicleManager
PaymentSystem
DatabaseHandler
Main Success Scenario
selectBookTicket()
displayAvailableDestinationsAndDates()
selectDestinationAndDate()
checkAvailableRoutes(destination, date)
alt [Routes available]
displayAvailableRoutes()
selectRoute()
displayAvailableVehicles(route)
selectVehicleAndSeats()
checkSeatAvailability(vehicle, seats)
alt [Seats available]
displayFare(route, vehicle, seats)
showBookingDetails(route, vehicle, seats, fare)
confirmBookingDetails()
showPaymentOptions()
alt [Pay Now (Online Payment)]
makePayment(fare)
paymentSuccess()
saveBooking(paymentType="Paid")
showConfirmation("Ticket Booked and Paid")
[Reserve and Pay on Spot]
saveReservation(paymentType="Cash on Spot")
showConfirmation("Ticket Reserved - Cash on Spot")
[No seats available]
showError("No seats available. Choose another vehicle.")
[No routes available]
showError("No routes available. Choose another date or destination.")
Extensions
alt [User cancels booking before confirmation]
cancelBooking()
returnToMainMenu()
alt [Database issues during reservation]
saveReservation()
reservationFailed()
showError("Reservation failed. Try again later.")
User
FASTTransportSystem
RouteManager
VehicleManager
PaymentSystem
DatabaseHandler

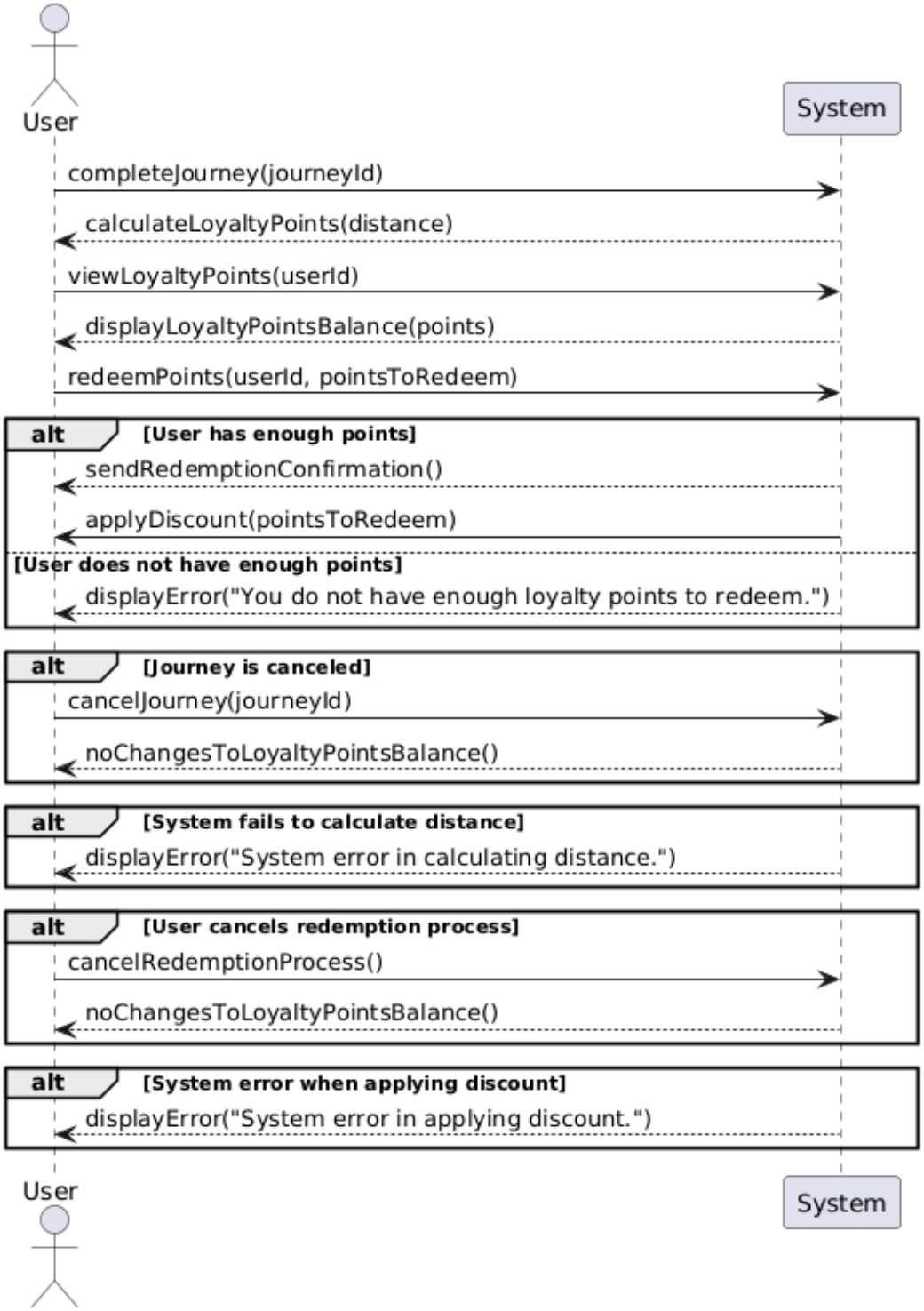
UC-03:


User
FASTTransportSystem
PaymentGateway
3rdPartyApp
DatabaseHandler
Main Success Scenario
completeBooking()
presentPaymentOptions(Cash, Bank/Debit Card)
alt [User selects Pay in Cash]
selectPayInCash()
reserveBooking(status="Pending - Cash on Spot")
displayConfirmation("Booking Reserved. Pay in Cash at Travel Time")
[User selects Bank/Debit Card]
selectBankDebitCard()
promptEnterCardDetails()
enterCardDetails()
processPayment(cardDetails)
alt [Payment Successful]
paymentSuccess()
updateBooking(status="Paid")
displayConfirmation("Payment Successful, Booking Confirmed")
[Payment Failed]
displayError("Payment failed. Try again or use a different method.")
retryOrCancelOption()
Extensions
alt [User cancels payment operation before confirmation]
cancelPayment()
returnToBookingScreen()
alt [Invalid bank card details]
displayError("Invalid card details. Please try again.")
promptEnterCardDetails()
alt [User fails to pay in cash by the designated time]
cancelBookingIfNotPaid()
displayNotification("Reservation canceled due to non-payment")
User
FASTTransportSystem
PaymentGateway
3rdPartyApp
DatabaseHandler

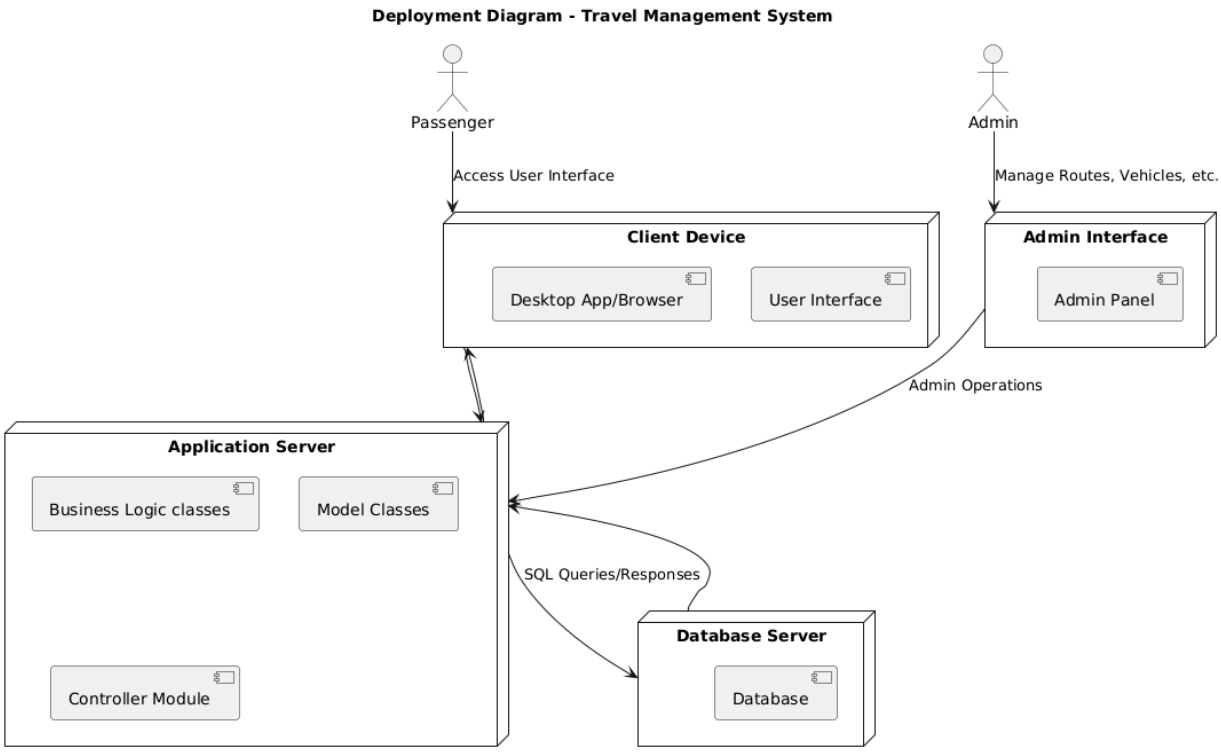
UC 04:


User
FASTTransportSystem
UserProfileManager
Validator
DatabaseHandler
Main Success Scenario
selectManageProfile()
initiateLogin()
validateCredentials(username, password)
credentialsValid()
alt
[Credentials valid]
showProfileInfo()
viewProfile()
updateProfileDetails(newDetails)
validateNewDetails(newDetails)
alt
[Details valid]
updateSuccess()
saveUpdatedDetails(newDetails)
[Details invalid]
showError("Invalid details")
[Credentials invalid]
showError("Invalid credentials")
User
FASTTransportSystem
UserProfileManager
Validator
DatabaseHandler

UC 05:


User
System
completeJourney(journeyId)
calculateLoyaltyPoints(distance)
viewLoyaltyPoints(userId)
displayLoyaltyPointsBalance(points)
redeemPoints(userId, pointsToRedeem)
alt
[User has enough points]
sendRedemptionConfirmation()
applyDiscount(pointsToRedeem)
[User does not have enough points]
displayError("You do not have enough loyalty points to redeem.")
alt
[Journey is canceled]
cancelJourney(journeyId)
noChangesToLoyaltyPointsBalance()
alt
[System fails to calculate distance]
displayError("System error in calculating distance.")
alt
[User cancels redemption process]
cancelRedemptionProcess()
noChangesToLoyaltyPointsBalance()
alt
[System error when applying discount]
displayError("System error in applying discount.")
User
System

2.4.4 Deployment diagram



3. Other Nonfunctional Requirements

3.1 Business Rules

The administrators can not be generated from within the program but have to be manually added from the database. The administrator will be directed to its own dashboard where it can perform their tasks and will not be able to access the user interface unless there is a customer account on a separate email and CNIC of that administrator. The users can only book for vehicles which travel on dates 7 days from the current date.

3.2 Operating Environment

1. **Hardware Specifications:**
    - **Processor:** At least an Intel Core i3 (or equivalent) processor is required.
    - **RAM:** A minimum of 4 GB is necessary, though 8 GB is recommended for optimal performance.
    - **Storage:** At least 500 MB of free disk space is needed for installing the application and its database.
2. **Operating System:**
    - **Supported Platforms:** Windows 10/11 (64-bit) or macOS 10.14 and above (if cross-platform functionality is implemented).
3. **Software Prerequisites:**
    - **Java Runtime Environment (JRE):** Requires JDK 11 or a later version compatible with JavaFX.
    - **Database Management System:** MySQL Community Server 8.0, configured to run on localhost.
    - **Development Tools:** IntelliJ IDEA, Eclipse, or NetBeans IDEs for development and debugging purposes.
4. **Database Configuration:**
    - **Setup:** MySQL database must be locally hosted with secure connection credentials stored in the application's configuration file.
    - **Port Settings:** Use the default MySQL port (3306) or a custom port configured by the user.
5. **Network Connectivity:**
    - **Local Access:** The JavaFX application should be able to connect to the MySQL server hosted on localhost via a local network.
6. **User Interface Requirements:**

- **Screen Resolution:** The application must support a minimum resolution of 750x475.
- **Input Methods:** Designed to work with a mouse and keyboard, with optional support for touch interaction.

7. **Additional Considerations:**
   - **Dependencies:** Requires JavaFX SDK and MySQL Connector/J for establishing database connectivity.
   - **Security:** Ensure the localhost database is secured with proper credentials to restrict unauthorized access.
   - **Testing Environment:** The application should be tested in a single-user setup using localhost configurations.